

(19)日本国特許庁 (J P)

(12)公開特許公報 (A)

(11)特許出願公開番号

特開平10-257060

(43)公開日 平成10年 (1998) 9月25日

(51)Int. Cl.⁶

識別記号

F I

H 0 4 L 12/28

H 0 4 L 11/20

G

H 0 4 Q 3/00

H 0 4 Q 3/00

審査請求 未請求 請求項の数11 O L (全 21 頁)

(21)出願番号 特願平9-52126

(22)出願日 平成9年 (1997) 3月6日

(71)出願人 000003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 高島 由彰

神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内

(72)発明者 伊瀬 恒太郎

神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内

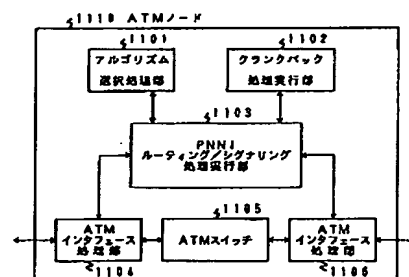
(74)代理人 弁理士 鈴江 武彦 (外6名)

(54)【発明の名称】 通信ノード、ATMノードおよびネットワークの経路選択接続方法

(57)【要約】

【課題】 PNNI プロトコルにおけるクランクバック処理による効果の受け方が、コネクションの種類により違ってしまうことによる不公平を解消すること。

【解決手段】 通信経路上のリソースを予め選択及び予約して通信開始するネットワークに用いられるATMノードにおいて、PNNI プロトコルに従って動作すると共にコネクション設定時、ソースノードにより最初に選択された経路上へのコネクションの設定可否、または選択された代替経路上へのコネクションの設定可否を決定するシグナリング処理手段(SG)1103と、SGがコネクション設定不可と判断したとき、PNNIのクランクバック処理を実行するクランクバック処理手段1102と、SGの実行するコネクション設定可否決定法をクランクバック実行の有無に応じ変更する手段1101を備える。ATMノードにおけるシグナリング処理時に、シグナリング処理がクランクバックの後か前かによりコネクション設定可否決定のアルゴリズムを変更することで不公平性を是正する。



【特許請求の範囲】

【請求項1】 通信経路上のリソースを予め選択および予約して通信を開始し、最初に選択した通信経路上のリソースが予約できなかった場合に、代替経路の選択処理が実行されるネットワークに接続されたノード装置において、

判断基準の異なる判断手段を複数備え、前記通信経路が選択される毎に、その選択された通信経路上のリソースの予約が可能か否かを指定の前記判断手段を用いて判断する第1のシグナリング処理手段と、

該シグナリング処理手段が前記リソース予約不可と判断したとき、代替通信経路の選択処理を実行する代替経路選択手段と、

該代替経路選択手段の選択した通信経路上にリソースの予約が可能か否かを判断する第2のシグナリング処理手段と、

前記第1のシグナリング処理手段の用いる前記判断手段を最初の選択による通信経路のリソース予約可否判断か否かにより指定変更するアルリズム変更手段と、を備えることを特徴とする通信ノード。

【請求項2】 通信経路上のリソースを予め選択および予約して通信を開始するATMネットワークに用いられるATMノードにおいて、

PNNIプロトコルに従って動作すると共に、コネクション設定時に、ソースノードにより最初に選択された経路上へのコネクションの設定可否、または選択された代替経路上へのコネクションの設定可否を決定するシグナリング処理手段と、

該シグナリング処理手段がコネクション設定不可と判断した際に、PNNIにおけるクラックバック処理を実行するクラックバック処理手段と、

前記シグナリング処理手段において実行する前記コネクション設定可否の決定法をクラックバック実行の有無に応じて変更するコネクション設定可否判定アルリズム変更手段と、を有することを特徴とするATMノード。

【請求項3】 ネットワークに発生している呼のうち、所要の帯域を要求している呼に対して、前記アルリズム変更手段による変更を実行するアルリズム変更領域決定手段を有することを特徴とする、請求項1に記載の通信ノード。

【請求項4】 ネットワークに発生している呼のうち、所要の帯域を要求している呼に対して、前記コネクション設定可否判定アルリズム変更手段による変更を行なうアルリズム変更領域決定手段を有することを特徴とする、請求項2に記載のATMノード。

【請求項5】 前記アルリズム変更領域決定手段を有し、

前記シグナリング処理手段は、前記最初に選択した通信経路のコネクション設定は、発生した呼の要求帯域以上の帯域がリンク中に存在するとき、前記代替経路のコネ

クションの設定は、前記アルリズム変更領域決定手段の決定した所定値以上の帯域がリンク中に存在するとき、設定許可するものであることを特徴とする請求項1に記載の通信ノード。

【請求項6】 前記アルリズム変更領域決定手段を有し、

前記シグナリング処理手段は、

前記最初に選択した通信経路のコネクション設定は、発生した呼の要求帯域以上の帯域がリンク中に存在するとき、前記代替経路のコネクション設定は前記アルリズム変更領域決定手段の決定した所定値以上の帯域がリンク中に存在するとき、設定許可するものであることを特徴とする請求項2に記載のATMノード。

【請求項7】 自ノードもしくはネットワーク全体に発生している呼が要求している要求帯域情報を収集／記憶する帯域情報収集／記憶手段と、

該帯域情報収集／記憶手段に記憶されている要求帯域のうち、前記アルリズム変更手段もしくはコネクション設定可否判定アルリズム変更手段を実行する要求帯域を選択するアルリズム変更領域選択手段とを有し、

前記シグナリング処理手段は、前記最初に選択した通信経路のコネクション設定は、発生した呼の要求帯域以上の帯域がリンク中に存在するとき、前記代替経路のコネクション設定は前記アルリズム変更領域選択手段により決定された所定値以上の帯域がリンク中に存在するとき、設定許可するものであることを特徴とする請求項1に記載の通信ノード。

【請求項8】 自ノードもしくはネットワーク全体に発生している呼が要求している要求帯域情報を収集／記憶する帯域情報収集／記憶手段と、

該帯域情報収集／記憶手段の記憶した要求帯域のうち、前記アルリズム変更手段もしくはコネクション設定可否判定アルリズム変更手段の実行する要求帯域を選択するアルリズム変更領域選択手段とを有し、

前記シグナリング処理手段は、前記最初に選択した通信経路のコネクション設定は、発生した呼の要求帯域以上の帯域がリンク中に存在するとき、前記代替経路のコネクション設定は前記アルリズム変更領域選択手段の決定した所定値以上の帯域がリンク中に存在するとき、設定許可するものであることを特徴とする請求項2に記載のATMノード。

【請求項9】 前記帯域情報収集／記憶手段とを有し、前記シグナリング処理手段は、前記最初に選択した通信経路のコネクション設定は、発生した呼の要求帯域以上の帯域がリンク中に存在するとき、前記代替経路のコネクション設定は前記帯域情報収集／記憶手段の記憶した要求帯域のうち、最大値の要求帯域以上の帯域がリンク中に存在するとき、設定許可するものであることを特徴とする請求項1に記載の通信ノード。

【請求項10】 前記帯域情報収集／記憶手段とを有し、

前記シグナリング処理手段は、前記最初に選択した通信経路のコネクション設定は、発生した呼の要求帯域以上の帯域がリンク中に存在するとき、前記代替経路のコネクション設定は前記帯域情報収集／記憶手段の記憶した要求帯域のうち、最大値の要求帯域以上の帯域がリンク中に存在するとき、設定許可するものであることを特徴とする請求項2に記載のATMノード。

【請求項11】 通信経路上のリソースを予め選択および予約して通信を開始し、最初に選択した通信経路上のリソースが予約できなかった場合に、代替経路の選択処理が実行されるネットワークにおける経路選択接続方法において、

判断基準の異なる判断アルゴリズムを複数用意し、前記通信経路が選択される毎に、その選択された通信経路上のリソースの予約が可能か否かを指定の前記判断アルゴリズムを用いて判断し、

該判断において前記リソース予約不可のときは、代替通信経路の選択処理を行ない、この選択した通信経路上にリソースの予約が可能か否かを判断するようにし、

また、使用する前記判断アルゴリズムは前記最初の選択による通信経路のリソース予約可否判断か否かにより指定変更することを特徴とするネットワークの経路選択接続方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、PNNIインタフェース(Private Network Network Interface)を有するATMノードに関し、特にATMコネクションを設定する際のシグナリング処理において、PNNIで規定されたシグナリング方式を実行するとともに、クランクバック処理をも実行するATMノードおよびネットワークの経路選択接続方法に関する。

【0002】

【従来の技術】 ATM(Asynchronous Transfer Mode；非同期転送モード)はマルチメディア通信に適した高速広帯域通信網(B-ISDN)を実現するための基本方式として生み出されたものである。すなわち、デジタル通信サービス用のネットワークとして通信速度が64 kbpsのISDN(Integrate Services Digital Network；統合サービス・デジタル網)が実現され、音声の他に、ファクシミリやデータの通信を提供できるようになったが、これはあくまでもデジタル電話網をベースとしたシステムとして考えられてきたものであり、従って、通信を行うにあたっての速度は音声通信に合わせた64 kbpsを基本としたものであった。

【0003】 しかし、動画像なども伝送の対象とし、さらにはビデオ・オン・デマンドなどの高度なサービスを提供することを目指すマルチメディア通信を実現する必要性から、さらに高速で、しかも、高機能、高信頼性の

通信サービスとして登場したのがB-ISDNの構想であり、このB-ISDNの検討の中から生まれてきた技術がATMである。

【0004】 そして、近年ではATM通信方式に対応したネットワーク製品が数多く市場に登場するようになってきており、特に、ATMの実装標準を作成する団体であるATM-Forum(ATMフォーラム)において決定されたUNI(User Network Interface)標準であるUNI3.0やUNI3.1の標準に準拠したATMスイッチやATMインタフェースを持った製品が多く市場に登場して、実用に供されており、企業内のネットワークをATM-LANに置き換えるところが徐々に増えてきている。

【0005】 また、ATMスイッチノード間インタフェースとして、ATMフォーラムがPNNI(Private Network Network Interface)の仕様を1996年に決定し、このPNNI方式のインプリメントが多くATMスイッチベンダーによって開始されている。

【0006】 PNNIインタフェース仕様では、複数のATMスイッチを接続したネットワークを仮想的な階層構造のネットワークとして認識する。そして、この仮想的ネットワークにおいては、PNNIインタフェース仕様においてピアグループ(Peer Group)と呼ばれる、複数のスイッチノードからなるサブネットを一つの論理的なノードと認識することで、記憶すべきトポロジー情報の削減を図っている。

【0007】 すなわち、このPNNIインタフェース仕様においては、“ピアグループ”と呼ばれる複数のATMスイッチノードからなるサブネットを、一つの論理的なノードと認識することで、複数のATMスイッチを接続したネットワークを仮想的な階層構造のネットワークとして認識するようになっている。そして、このような仮想的な階層構造でトポロジー(網形態)を認識することで、PNNI方式においては、ネットワーク内の各ATMスイッチノードが記憶すべきトポロジー情報の削減を図っている。

【0008】 図8に、PNNI方式におけるネットワークトポロジーの認識方法を示す。図9に示した一例としてのATMネットワークにおいては、交換ノードである複数のATMスイッチノード101、102、～、105、106、～122が存在し、それらのATMスイッチノード間がリンクによって接続されている構成になっている。

【0009】 PNNI方式では、これらのATMスイッチノード間で、各ATMスイッチノード間のリンクや論理的ATMスイッチノードのトポロジー情報の授受を行っている。図8のATMネットワークの例においては、上記の22個の実存ATMスイッチノード101、～122を6つのATMスイッチノード群(ピアグルー

ブ)に分割し、それぞれのピアグループを論理的なATMスイッチノード31~36と認識するようになっている。

[0010]ここで実存ATMスイッチノード101, 102, 103, 104からなるピアグループが論理的なATMスイッチノード31、実存ATMスイッチノード105, 106, 107からなるピアグループが論理的なATMスイッチノード32、ATMスイッチノード108, 109, 110, 111からなるピアグループが論理的なATMスイッチノード33、実存ATMスイッチノード112, 113, 114, 115からなるピアグループが論理的なATMスイッチノード34、実存ATMスイッチノード116, 117, 118からなるピアグループが論理的なATMスイッチノード35、実存ATMスイッチノード119, 120, 121, 122からなるピアグループが論理的なATMスイッチノード36である。

[0011]よって、PNNIにおいては、図8の下部に示した22個の実在するATMスイッチノード101, ~122によって構成されるネットワークを、仮想的な“下位階層のネットワーク”として認識する。また、図8の上部に示した6個の論理的なATMスイッチノード31, ~36によって構成される論理的なネットワークを、仮想的な“上位階層のネットワーク”と認識する。

[0012]また、ATMネットワークにおいては、複数種類のアプリケーションを同一のデータリンクプロトコルによって転送するために、いくつかのQOS (Quality of Service; サービス品質) パラメータが設定されており、これらの値をコネクション設定要求と同時にネットワークに対して要求することができるようになっている。

[0013]QOSパラメータには有効帯域、転送遅延時間、転送遅延揺らぎなど複数種類のパラメータがあり、これらの複数パラメータを最適化するルーティング経路を求める。

[0014]実存ATMスイッチノード101から実存ATMスイッチノード122にデータ伝送する場合を例に、図を用いて具体的に説明する。この場合、発呼ノードとなる図8に示したネットワーク上の実存ATMスイッチノード101は次のように振る舞う。すなわち、PNNI方式の場合、実存ATMスイッチノード101は前述のような仮想的階層構成のトポロジー認識方式を用いて、ネットワーク全体を図9のようなトポロジーとして認識することになる。

[0015]図9の例は、認識したネットワークのQOSパラメータとして、“遅延時間”と“有効帯域”が与えられた場合を示している。つまり、図9においては、例えば、発呼ノードであるスイッチノード101から隣接のスイッチノード102間においては、“遅延時間”

は“1msec”で“有効帯域”は“10Mbps”であり、発呼ノードであるスイッチノード101から隣接のスイッチノード103間においては、“遅延時間”は“5msec”で“有効帯域”は“30Mbps”であり、スイッチノード102から隣接のスイッチノード104間においては、“遅延時間”は“2msec”で“有効帯域”は“20Mbps”であり、スイッチノード103から隣接のスイッチノード104間においては、“遅延時間”は“3msec”で“有効帯域”は“40Mbps”であり、スイッチノード104から隣接のスイッチノード32間においては、“遅延時間”は“10msec”で“有効帯域”は“50Mbps”であり、スイッチノード32から隣接のスイッチノード33間においては、“遅延時間”は“10msec”で“有効帯域”は“20Mbps”であり、…といった具合であることが示されている。

[0016]ここで、ATMスイッチノード101からATMコネクションを設定する場合には、まず、呼を発生した実存ATMスイッチノード101において、図9に示したトポロジーのネットワークを用いて、ATMコネクションの設定経路を探索し、図9に示したトポロジー上でのコネクション設定経路を決定する。

[0017]つまり、呼を発生した実存ATMスイッチノード101は、自己の所属するピアグループ内における他の実存ATMスイッチノード102, 103, 104および仮想的上位階層ネットワークでの論理的なATMスイッチノードである論理的ATMスイッチノード32, 33, 34, 35の中のいくつかのノードを経て最終的に論理的ATMスイッチノード36に辿り着くことのできる1つの経路を探し、そのルートでのQOS (サービス品質) パラメータからその探したその経路が適正であるか否かを判断する。

[0018]その結果、不適正であれば、別経路を探して同様に、経路が適正であるか否かを判断し、適正であればこれを設定経路候補とし、次の動作であるコネクション設定可否の判定処理に移る。

[0019]コネクション設定可否の判定処理は次のようにして行う。実存ATMスイッチノード101は、その設定経路上にDTLスタックと呼ばれる経路情報を書き込んだパケットを送出し、各リンク上にコネクションが設定可能であるかどうかを順次判断していく。この時、図9上の論理的なATMスイッチノードにおいては、各論理的なスイッチノード内の仮想的に下位の階層のATMスイッチノードのトポロジーを用いて、各ピアグループ内のコネクション設定経路を独自に選択し、その選択したコネクション設定経路上にDTLスタックと呼ばれる経路情報を書き込んだパケットを転送し、各リンク上にコネクションが設定可能であるかどうかを順次判断していく。

[0020]よって、図9に示したトポロジー中の論理

的ATMスイッチノード32においては、自己のピアグループ所属の実存ATMスイッチノードである図8に示したネットワーク中の実存ATMスイッチノード105, 106, 107間でのコネクション設定経路を決定し、同様に、図9に示したトポロジー中の論理的ATMスイッチノード36においては、自己のピアグループ所属の実存ATMスイッチノードである図8に示したネットワーク中の実存ATMスイッチノード119, 120, 121, 122間でのコネクション設定経路を決定することになる。

【0021】このようにして、PNNIインタフェース仕様においては、“ピアグループ”と呼ばれる複数のATMスイッチノードからなるサブネットを、一つの論理的なノードと認識することで、複数のATMスイッチを接続したネットワークを仮想的な階層構造のネットワークとして認識し、そして、このような仮想的な階層構造でトポロジー（網形態）を認識することで、ネットワーク内の各ATMスイッチノードが記憶すべきトポロジー情報の削減を図っている。

【0022】また、PNNI方式においては、コネクションを設定する際の呼設定処理を、いくつかの段階に分けて規定している。

【0023】まず、ソースノード（すなわち、コネクションを設定しようとした端末を収容しているATMスイッチノード）において、コネクション全体の経路の選択／決定をするソースルーティング処理が実行される。そして、その結果、選択された経路上に“DTLスタック”と呼ばれる経路情報を書き込んだスタックを転送してゆき、各リンク上のリソースを確保していく処理が実行される。リソースが確保できれば、その確保したリソースを用いて発呼側と着呼側の端末間で通信が実施されることになる。

【0024】更にこのリソース確保に失敗した場合には、“クランクバック（Crankback）”と呼ばれる代替経路選択処理が実行され、これによって、前述の“DTLスタック”を適宜なるノードまで差し戻した後、新たなコネクションの経路の選択／決定処理を実行する。

【0025】PNNIの仕様においては、このようなクランクバック処理によって、ネットワーク全体での呼損率の改善を図るようにしている。

【0026】以下に、図10のようにトポロジーが認識されているネットワークを用いて、PNNIプロトコルにおけるDTLスタックを用いたシグナリング手順と、その時に実行されるクランクバック手順について説明する。

【0027】[DTLスタックによるシグナリング] PNNIルーティングでは、DTLスタックと呼ばれる、発呼ノードにおいて選択された経路情報を記憶しているスタックを用いたシグナリング方式が示されている。D

TLスタックを用いたルーティングの手順は以下のようになっている。

【0028】[段階1]. 発呼ノード（発呼端末の収容されているATMスイッチノード）でルーティングを実行し、その選択経路を書き込んだDTLスタックを作成する。

【0029】[段階2]. この作成したDTLスタックを、選択した経路上に転送していく。

【0030】[段階3]. 転送されたDTLスタックが、上記選択した経路上のピアグループに達すると、そのDTLスタックに書き込まれている情報を基に、そのピアグループにおいてグループ内の経路選択を行なう。

【0031】[段階4]. 到着したDTLスタックの経路情報の一部を、上記新たにピアグループ内で選択された経路情報に書き換え、この一部書き換えた新たなDTLスタックを選択経路上に転送する。

【0032】[段階5]. 上述の手順[段階2]. ～[段階4]. の処理を順次行ない、DTLスタックが着呼ノード（着呼先の端末が収容されているATMスイッチノード）に到着したところで、ルーティングが終了する。

【0033】このように、PNNIルーティングでは、選択した経路情報を記憶したDTLスタックを当該選択した経路に転送し、この経路上のピアグループに到達するとピアグループではグループ内の経路を選択してこのDTLスタックの経路情報の一部を当該選択した経路情報に書き換え、この書き換えられたDTLスタックを当該DTLスタックの経路情報に従った経路上に転送するといったことを繰り返して最終的に着呼先に送るようにする。

【0034】具体的に例を示す。

【0035】図10において、実存の階層である下位階層においてその実ノード“A. 1. 1”から実ノード“C. 2. 2”に対して呼が発生した場合を考えると、実ノード“A. 1. 1”（ATMスイッチによるノード）は、自ノードが属しているピアグループ以外は、全て論理ノードとして捉えるので、ネットワークを図11(a)のような論理的トポロジーであると認識している。

【0036】そして、この論理ネットワーク上で経路選択アルゴリズムを実行し、発呼ノードである“A. 1. 1”から、着呼ノードである“C. 2. 2”（図11(a)では論理ノードC）への経路を選択する。

【0037】ここで、発呼ノード“A. 1. 1”が選択した選択経路が、発呼ノードである実ノード“A. 1. 1”→実ノード“A. 1. 2”→実ノード“A. 1. 3”→ピアグループ“A. 2”（論理ノード）→ピアグループ“A. 3”（論理ノード）→ピアグループ“B”（論理ノード）→ピアグループ“C”（論理ノード）であったとすると、その時に、作成されるDTLスタック（経路情報を書き込んだパケット）は図11(b)のよ

うになる。

【0038】このように作成されたDTLスタックは、発呼ノード“A. 1. 1”で選択された上記の経路上を転送され、各ピアグループで、以下のようにDTLスタックの書換えが行なわれる(図12参照)。

【0039】(1). 図12(b)に示す如く、ピアグループ“A. 1. 3”から伝送された[A. 1. 1 A. 1. 2 A. 1. 3][A. 1-A. 2-A. 3][A-B-C]なる内容のDTLスタック

“[1.]”がピアグループ“A. 2”に到着する。するとこのDTLスタック“[1.]”が当該ピアグループ“A. 2”に到着した時点で、このピアグループ

“A. 2”内の経路選択が行なわれる。そして、その結果、“A. 2”内の経路として、[A. 2. 2-A. 2. 1]が選択されたとすると、DTLスタック

“[1.]”の“A. 1”内経路情報部分が、新たに選択された当該[A. 2. 2-A. 2. 1]なる内容の情報に書き換えられ、図12(b)に示す如く、[A. 2. 2-A. 2. 1][A. 1-A. 2-A. 3][A-B-C]なる内容のDTLスタックとなる。そして、

DTLスタック“[1.]”の[A. 2. 2-A. 2. 1]なる内容の情報に従って、“A. 2. 2”→“A. 2. 1”と伝送される。“A. 2. 1”ではDTLスタック“[1.]”の[A. 1-A. 2-A. 3]なる情報より、“A. 3”のピアグループにこのDTLスタック“[1.]”を送ることになるが、“[1.]”がピアグループ“A. 3”に到着した時点で、このピアグループ“A. 3”内の経路選択が行なわれる。そして、その結果、“A. 3”内の経路として、[A. 3. 1-A. 3. 2-A. 3. 3]が選択されたとすると、DTLスタック“[1.]”の“A. 3”内経路情報部分が、新たに選択された当該[A. 3. 1-A. 3. 2-A. 3. 3]なる内容の情報に書き換えられ、図12(c)に示す如く、[A. 3. 1-A. 3. 2-A. 3. 3][A. 1-A. 2-A. 3][A-B-C]なる内容のDTLスタック“[2.]”となる。そして、DTLスタック“[2.]”の[A. 3. 1-A. 3. 2-A. 3. 3]なる内容の情報に従って、“A. 3. 1”→“A. 3. 2”→“A. 3. 3”と伝送される。そして、DTLスタック“[2.]”の[A-B-C]なる内容の情報に従って、DTLスタック“[2.]”は、“B”のピアグループに伝送される。

【0040】(2). 次に、このDTLスタック

“[2.]”がピアグループ“B”に到着した時点で、ピアグループ“B”内の経路選択が行なわれる。“B”内の経路として、[B. 1. 1-B. 1. 3]が選択されたとすると、DTLスタック“[2.]”の、[A. 3. 1 A. 3. 2 A. 3. 3]なる内経路情報部分が、図12(c)に示すように新たに選択された

[B. 1. 1-B. 1. 3]なる内経路情報に書き換え

られる。

【0041】このようにして、発呼ノードで作成されたDTLスタックは、経路情報に従って該当ピアグループに転送され、ピアグループでは自グループ内の経路選択を行って、DTLスタックの該当部分を書き換えるといった処理をしてゆく。そして、これによって転送経路となったノードでのリソースが確保される。

【0042】選択された転送経路の途中段階でリソース確保に失敗すれば、代替経路選択処理(クランクバック)が行われ、前述のDTLスタックを適宜なるノードまで差し戻してから、再び、新たなコネクションの経路の選択/決定処理を実行する。

【0043】このクランクバックのアルゴリズムは次の通りである。

【0044】[クランクバックアルゴリズム] PNNIでは、トポロジー情報を周期的に書き換えた後に、そのトポロジー情報を各ノードに通知するようになっている。よって、書き換え周期間に発生した呼によって生じたトポロジー情報の変化は、次の書き換えが終了するまで他のノードには通知されないことになる。このため、トポロジー情報の不一致による呼損が発生することが予想される。

【0045】この問題に対し、PNNIでは“クランクバック”と呼ばれる、呼の要求するコネクションが設定できなかった場合に、他のコネクション設定可能な経路を順次探索するアルゴリズムを採用している。

【0046】具体的には、発呼ノードで選択した経路上にコネクションが設定不可能であった場合に、前のピアグループ(すなわち、DTLスタックを書き直したピアグループ)に戻って、もう一度経路選択をやり直す方式である。

【0047】図11に示した論理ネットワーク上のルーティングにおいて、クランクバックが発生した場合の、その前後で変更されるコネクション経路とDTLスタックの一例を図13に示し、その実行手順を以下に示す。

【0048】発呼端末が収容されたノードである発呼ノードを“A. 1. 1”、着呼先の端末が収容されたノードをピアグループ“C”として話を進める。発呼ノード“A. 1. 1”で選択された経路の情報であるDTLスタックの内容が図13(a)に示す如く、[A. 1. 1 A. 1. 2 A. 1. 3][A. 1-A. 2-A. 3][A-B-C]であったとする。この場合、DTLスタックは“A. 1. 1”→“A. 1. 2”→

“A. 1. 3”の経路を辿るように転送される、さらに“A. 1. 3”からピアグループ“A. 2”へ、そして、“A. 2”から“A. 3”へと転送されることになる。

【0049】1. 今、発呼ノード“A. 1. 1”で選択された経路上をDTLスタックが転送され、論理ノード“A. 2”～“A. 3”間に到着したところで、コネ

クション設定不可能であったとする(Reject)。

【0050】2. すると“A. 2”～“A. 3”間にコネクション設定ができなかったことを経路を逆戻りしながら通知していく(“Reject通知”)。ここでは、“A. 2”で新たな経路が選択できないので、発呼ノード“A. 1. 1”までクランクバックされる。

【0051】3. “Reject通知”を受け取ったノード“A. 1. 1”は、論理ノード“A. 2”～“A. 3”間のリンクは無いものとして経路選択を実行する。そして、選択し直した経路情報を格納したDTLスタックを新たに作成し、新たに選択した経路上に転送していく。

【0052】このように、クランクバックアルゴリズムにおいては、コネクションが設定できなかった場合に、適宜なところに戻って、新たなDTLスタックを作成することになる。よって、クランクバックを実行するためには、各ピアグループやノードが、自ピアグループや自ノードを通過したDTLスタックを記憶しておかなければならない。

【0053】このようなクランクバック処理による効果としては、ネットワーク全体の呼損率の改善が上げられる。しかし、この効果は、ネットワーク中に発生する全てのコネクションに公平に与えられるとは限らないことがわかっていてる。

【0054】例えば、複数種類の帯域を要求するコネクションが発生する、いわゆるマルチコネクションの環境においては、クランクバックの効果は、その複数コネクションの中の要求帯域の小さなコネクションを中心に与えられることになる。また、クランクバックを用いたことによって、要求帯域の大きなコネクションは、逆に呼損率が劣化してしまう場合もあることが知られている。

【0055】これはマルチコネクションの環境における問題点として、従来から指摘されていた“コネクションの種類の違いによる呼損率の不公平性”という問題点を助長する結果を招いていることになる。

【0056】通常、コネクションの要求する帯域の違いによる呼損率の不公平性は、リンクの帯域と要求帯域の関係(いわゆる分割損)によって決まるものであり、ある意味において仕方のないものである。

【0057】しかし、このようなクランクバックという新たな機能を追加したことによって、この不公平性が助長されることは、ネットワーク中に発生するコネクションを同等に扱うことができないということを意味している。これは、ユーザーから見るとサービス提供が公平になされていないように感じる。

【0058】

【発明が解決しようとする課題】PNNI方式においては、コネクションを設定するに際してネットワーク全体の呼損率の改善をはかるため、リソースの確保に失敗すると、クランクバック処理が行われるが、この効果は、

ネットワーク中に発生する全てのコネクションに公平に与えられるとは限らない。

【0059】例えば、複数種類の帯域を要求するコネクションが発生するいわゆるマルチコネクションの環境においては、クランクバックの効果は、その複数コネクションの中の要求帯域の小さなコネクションを中心に与えられる。従って、要求帯域の大きなコネクションは、クランクバックにより、逆に呼損率が劣化してしまう場合もある。

10 【0060】これはマルチコネクションの環境における問題点として、従来から指摘されていた“コネクションの種類の違いによる呼損率の不公平性”を顕在化させる結果を招いている。

【0061】このように、現在規定されているクランクバック処理方式では、ネットワーク全体での呼損率を改善することはできるが、個々のコネクションの単位で見ると、その効果が公平に与えられてはいないという問題点があった。故にこの不公平を改善する技術の確立が要望される。

20 【0062】そこで、この発明の目的とするところは、マルチコネクションの環境においてPNNIのクランクバックアルゴリズムを実行したとしても、そのクランクバックの効果は、全てのコネクションが公平に享受できるようにし、ネットワーク中に発生する各コネクションの呼損率を均等に改善することのできる呼設定アルゴリズムを実行するATMノードを提供することにある。

30 【0063】また、通信経路上のリソースを予め選択および予約して通信を開始し、最初に選択した通信経路のリソースが予約できなかった場合に代替経路の選択処理を行うネットワークにおける経路選択接続方法を提供することにある。

【0064】

【課題を解決するための手段】上記目的を達成するため、本発明は次のように構成する。

40 【0065】第1には、通信経路上のリソースを予め選択および予約して通信を開始するネットワークであって、最初に選択した通信経路上のリソースが予約できなかった場合に、代わりとなる新たな代替経路の選択処理が実行されるネットワークにおいて、通信ノードは、通信ネットワークに発生した呼の要求に従って通信を行なうための、通信ネットワーク内での通信経路選択手段と、その選択経路上での通信リソースの獲得を行なう通信リソース獲得手段を有する。また、前記リソース獲得手段で選択された通信経路上に通信リソースが獲得できなかった場合に、再度、通信経路の選択をやり直す通信経路再選択手段と、その再選択された通信経路上での通信リソースの獲得を行なう通信リソース再獲得手段を有する。さらに、前記通信リソース獲得手段と前記通信リソース再獲得手段において実行する通信リソースの獲得
50 アルゴリズムを変更するアルゴリズム変更手段を有して

いる。

【0066】これにより、前記通信リソース獲得手段によって通信ネットワーク中に発生する呼が受ける影響と、前記通信リソース再獲得手段によって通信ネットワーク中に発生する呼が受ける影響を独立のものとすることができ、前記通信経路選択／再選択手段によって呼が受ける影響を分散することができる。

【0067】第2には、前記ネットワークのATMノードは、クランクバック手段を含めた、PNNIプロトコルを実行するためのPNNIプロトコル実行手段と、コネクション設定時に、ソースノードによって最初に選択された経路やクランクバック実行後の代替経路選択処理によって選択された経路上へのコネクションの設定可否を決定するシグナリング処理手段とを有する。また、該シグナリング手段において、ソースノードによって最初に選択された経路上へのコネクションの設定可否を決定するのか、クランクバック実行後の代替経路選択処理によって選択された経路上へのコネクションの設定可否を決定するのかを認識するシグナリング処理認識手段と、該シグナリング処理認識手段によって認識されたシグナリング処理において実行される、コネクションの設定可否を決定するアルゴリズムを変更するコネクション設定可否判定アルゴリズム変更手段とを有している。

【0068】これによって、ネットワーク中に発生する呼が、最初のソースノードにおけるシグナリング処理によって受ける影響と、クランクバック処理の後の代替経路へのシグナリング処理によって受ける影響を分散することができる。

【0069】第3には、上記第1または第2の構成において、ネットワークに発生している呼の中の、適当な帯域を要求している呼に対してのみ、前記アルゴリズム変更手段もしくはコネクション設定可否判定アルゴリズム変更手段を実行するためのアルゴリズム変更領域決定手段を備える構成とした。

【0070】第4には、通信ノードおよびATMノードは、第1の構成または第2の構成の通信ノードおよびATMノードが有する手段に加えて、前記シグナリング処理認識手段によって、実行されているシグナリング処理がソースノードによって最初に選択された経路上へのコネクションの設定可否を決定していると認識された場合には、発生した呼が要求する帯域以上の帯域がリンク中に存在すれば、コネクションの設定を許可し、前記シグナリング処理認識手段によって、実行されているシグナリング処理がクランクバック処理の実行後に代替経路として選択された経路上へのコネクションの設定可否を決定していると認識された場合には、何らかの方法によって定められた、ある一定以上の帯域がリンク中に存在するならば、コネクションの設定を許可するコネクション設定可否判定アルゴリズム変更手段を有している。

【0071】これによって、ネットワーク中に発生する

呼が、その要求する帯域の値に依存せずに、代替経路選択処理やクランクバック処理による効果を受けることができるようになる。

【0072】第5には、第1の構成または第2の構成の通信ノードおよびATMノードが有する手段に加えて、前記シグナリング処理認識手段によって、実行されているシグナリング処理が、ソースノードによって最初に選択された経路上へのコネクションの設定可否を決定していると認識された場合には、発生した呼が要求する帯域以上の帯域がリンク中に存在すれば、コネクションの設定を許可し、前記シグナリング処理認識手段によって、実行されているシグナリング処理がクランクバック処理の実行後に代替経路として選択された経路上へのコネクションの設定可否を決定していると認識された場合には、ネットワーク中に発生しているコネクション中で、最も大きな帯域を要求している呼が要求する帯域以上の帯域がリンク中に存在すればコネクションの設定を許可する、コネクション設定可否判定アルゴリズム変更手段を有している。

【0073】これによって、ネットワーク中に発生する全ての呼が、その要求する帯域の値に依存せずに、代替経路選択処理やクランクバック処理による効果を受けることができるようになる。

【0074】また、本発明は、通信経路上のリソースを予め選択および予約して通信を開始し、最初に選択した通信経路上のリソースが予約できなかった場合に、代替経路の選択処理が実行されるネットワークにおける経路選択接続方法において、判断基準の異なる判断アルゴリズムを複数用意し、前記通信経路が選択される毎に、その選択された通信経路上のリソースの予約が可能か否かを指定の前記判断アルゴリズムを用いて判断し、該判断において前記リソース予約不可のときは、代替通信経路の選択処理を行ない、この選択した通信経路上にリソースの予約が可能か否かを判断するようにし、また、使用する前記判断アルゴリズムは前記最初の選択による通信経路のリソース予約可否判断か否かにより指定変更することを特徴とする。

【0075】これによって、ネットワーク中に発生する呼が、その要求する帯域の値に依存せずに、代替経路選択処理やクランクバック処理による効果を受けることができるようになる。

【0076】

【発明の実施の形態】本発明は、PNNIプロトコルを実装したATMノードによって構成されるネットワークにおいて、シグナリング処理がクランクバック後の処理であるのか、クランクバック前の処理であるのかによって、コネクション設定可否を決定するアルゴリズムを変更するようにし、これによって、クランクバック処理による効果の受け方が、コネクションの種類によって不公平にならないようにするもので、以下、図面を参照しな

がら、本発明の実施の形態について詳細に説明する。

【0077】ATMノードを用いた本発明の適用されるATMネットワークのトポロジーの一例を図1に示す。図1において、101~116はそれぞれATMノード(ATMスイッチノード)であり、201、202、203、204はピアグループである。ピアグループ201には101~104のATMノードが所属し、ピアグループ202には105~108のATMノードが所属し、ピアグループ203には109~111のATMノードが所属し、ピアグループ204には113~116のATMノードが所属する。

【0078】このようなATMネットワークにおいて、以下の実施の形態においては、ATMノード101から、ATMノード116に対し、PNNIアルゴリズムを用いてコネクションを設定する際の処理について述べ、その際に実行されるルーティング/シグナリング処理および、本発明のATMノードによるクランクバック処理の詳細について説明する。

【0079】まず、図1中のATMノード101がPNNIにおけるトポロジー認識方式によって認識する仮想的なネットワークトポロジーの一例を、図2に示す。

【0080】図2に示すように、ATMノード101は、ネットワーク中のATMノード101が属している以外のピアグループを論理的なATMノードとして認識している(図1中の202、203、204)。そして、この図2のトポロジーを用いて、ATMノード101での最初のルーティング処理が実行される。

【0081】図3に、本例におけるPNNIルーティング/シグナリング処理のフローチャートの一例を示し、図4に、本発明のATMノードにおいて実行される呼設定可否を判定する際のアルゴリズムの一例を示す。

【0082】本発明のATMノードにおいて実行される判定アルゴリズムは、実行される呼設定の可否の判定が、“ソースノードにおいて最初に選択された経路上へのコネクション設定の可否判定”であるのか、“クランクバック処理の後に選択された経路上へのコネクション設定の可否判定”であるのかによって、実行するアルゴリズムを変更するという点に特徴を持つものである。詳細は後述する。

【0083】以下に、図1、図2のネットワークトポロジーと、図3のフローチャートと図4の実行アルゴリズムの一例を用い、本発明のATMノードを用いた場合のPNNIアルゴリズムによるATMコネクションの設定処理(ルーティング/シグナリング処理)の一例を示していく。

【0084】本発明のATMコネクション設定処理は、図3に示す如きであり、発呼端末を収容した発呼ノードにおいて、発呼端末からのコネクション設定要求を受け取ると、発呼ノードでは経路選択し、その経路選択した経路の情報をDTLスタックに記憶してその選択した経

路に転送し、ピアグループでは転送されてきたDTLスタックの経路情報から自ピアグループにおいてその経路を辿ってリンク(接続)していくことが可能か否かを判断して、可能ならば、自ピアグループでの経路を選択して、受け取ったDTLスタックの情報の一部をこの選択した経路の情報に変更し、その経路情報に従った経路へと、このDTLスタックを転送し、また、DTLスタックの経路情報の経路を用いてのリンクができないならば、代替経路があるか否かを調べ、代替経路があれば、自ピアグループでの代替経路を選択して、DTLスタックの情報の一部をこの選択した経路の情報に変更し、その経路情報に従った経路へと、このDTLスタックを転送し、また、DTLスタックの経路情報の経路を用いてのリンクができず、しかも、代替経路もないならば、代替経路を選べる位置のノードまでDTLスタックを戻し、再び、経路選択をし直すといったことを行う(クランクバック)。

【0085】ただし、クランクバックを行って、代替経路の選択を行う場合には、コネクション設定可否の判断に用いるアルゴリズムを変更する。具体的には、ネットワーク内に発生している呼が要求している帯域の中の最も大きな帯域と、コネクションを設定しようとしている選択経路上のリンクやATMノード内の有効帯域とを比較して、コネクションの設定可否の判断をする。

【0086】なお、クランクバックを行っていない場合のコネクション設定可否の判定アルゴリズムは、送られてきたDTLスタックに記されている要求帯域(Band, Req)がコネクションを設定しようとしているリンクやATMノード内の有効帯域(Band, Remain)よりも小さかった場合に、要求されたコネクションを設定可能と判断し、逆に大きい若しくは等しい場合には、コネクションの設定を不可能と判断する。

【0087】コネクション設定可否の判断、すなわち、リンク(接続)していくことが可能か否かの判断は図4の流れに沿う。

【0088】このように、ピアグループにおいて、DTLスタックの情報に基づいてコネクション設定可否を行うに当たり、クランクバックによる処理を行った後であるか、否かによりコネクション設定可否判定のアルゴリズムを変えるようにしたことで、クランクバック処理を行う前の判定条件と、クランクバックを行った後の判定条件を独立にでき、クランクバック処理を行った後の判定条件を全ての呼に平等に課せられるようにしたので、クランクバック処理による効果の受け方がコネクションの種類にかかわらず公平になる。

【0089】以上が、本発明の概要である。つぎに処理の具体的内容について、詳述する。はじめに、図3の流れを説明する。コネクション設定要求(呼設定要求)を受けると処理S701を実行する。

【0090】[図3の処理S701] まず、ATMノード

ド“101”に收容されているATM端末から、呼設定要求が送られてくると、ATMノード“101”は、その要求情報に記されている宛先端末を收容しているATNノード（ここではATMノード“116”とする）に対してコネクションを設定するための経路探索処理を実行する（図3の処理S701）。

【0091】このATMノード“101”によるルーティング処理によって、経路[101-103-104-203-204]が選択されたとなると、ソースノードであるATMノード“101”は、上記の選択経路を書き込んだDTLスタックを作成し、選択した経路に沿って、そのDTLスタックを順次転送していくことになる。この時にソースノードであるATMノード“101”は、作成したDTLスタックをコネクションのセットアップ要求メッセージ（いわゆるSetupメッセージ）に記述して転送していく。

【0092】以下においては、このメッセージのことを“DTLスタック（Setupメッセージ）”と記述していくものとする。このDTLスタック（Setupメッセージ）の転送に際し、選択された経路上に存在するATMノードは、発生した呼が要求しているQOS情報と、自ATMノードや経路上の隣接リンクのリソース状況を比較し、要求されているQOS情報を満足することができる場合に、要求されたコネクションの設定が可能と判断し、選択された経路上の次のATMノードにDTLスタック（Setupメッセージ）を転送する。コネクション設定可否の判定は図4のアルゴリズムに従う。

【0093】本実施例においては、要求されるQOS情報として、帯域情報を用いてコネクションの設定可否を判定していく場合を考える。

【0094】【図3の処理S710】S710での処理は次の通りである。前述の処理（S701での処理）の後にDTLスタック（Setupメッセージ）を受け取った各ATMノードにおいて、実際に要求された帯域が、自ノード内とノード間のリンク上に設定可能であるかを判断する。

【0095】この時の、各ATMノードで具体的に実行されるアルゴリズムの一例を図4に示している。各ATMノードでは、まず、送られてきたコネクション設定要求が、ソースノードにおける最初のルーティング処理の結果、作成されたDTLスタック（Setupメッセージ）によって送られてきたものか、それともクラックバック処理の後に作成されたDTLスタック（Setupメッセージ）によって送られてきたものかを判断する（S801）。

【0096】この時の、送られてきたDTLスタック（Setupメッセージ）の種別は、シグナリングメッセージ（Setupメッセージ）中のクラックバック・インフォメーション・エレメント（Crankback Information Element）の記述を

参照することによって可能である。

【0097】さらに、このクラックバック・インフォメーション・エレメントに記述されているコース・フィールド（Cause Field）を参照することによって、クラックバック処理が実行された理由を知ることができるので、そのCause Fieldの値によって実行する処理を変更することも可能である。ただし、本実施例では、クラックバック処理は、呼の要求する帯域が実際に存在しなかった場合にのみ発生したものと仮定し、このCause Fieldの値による処理の変更方法については省略する。

【0098】図5、図6に、クラックバック・インフォメーション・エレメント（Crankback Information Element）とコース・フィールド（Cause Field）の一例を示す。

【0099】各ATMノードは、上記の方法によって、送られてきたDTLスタック（Setupメッセージ）がソースノード（ATMノード101）における最初のルーティング処理の結果作成されたものか、それともクラックバック処理の後に、新たに作成されたものかの判断ができる（図4の処理S801）。

【0100】この結果、送られてきたDTLスタック（Setupメッセージ）がソースノードにおける最初のルーティング処理の結果作成されたものであった場合には、通常の呼設定可否の判定アルゴリズム（図4のS803の処理“アルゴリズム-2”）を実行する。図4では、要求されたコネクションの種別がCBRコネクションであると仮定し、要求帯域をピークアサイン処理により、設定可能かどうかの判断をする例を示している（図4の処理S803）。

【0101】つまり、送られてきたDTLスタック（Setupメッセージ）に記されている要求帯域（図4ではBand. Req）が、コネクションを設定しようとしている選択経路上のリンクやATMノード内の有効帯域（図4のBand. Remain）よりも小さかった場合に、要求されたコネクションを設定可能と判断し、逆に大きい／等しい場合には、コネクションの設定を不可能と判断する。

【0102】また、送られてきたDTLスタック（Setupメッセージ）がクラックバック処理の後のルーティング処理の結果作成されたものであった場合には、ネットワーク内に発生している呼が要求している帯域の中の最も大きな帯域と、コネクションを設定しようとしている選択経路上のリンクやATMノード内の有効帯域とを比較して、コネクションの設定可否の判断をするようになっている（図4のS802の処理“アルゴリズム-1”）。

【0103】この設定可否の判断処理においても、要求されたコネクションの種別がCBRコネクションであると考え、ピークアサイン処理によって、そのコネクショ

ン設定の可否を判断をする例を示している。

【0104】具体的には、本実施例のATMノード内に、ネットワーク内に発生している呼が要求している帯域情報を記憶する手段を設け、その手段に記憶されている要求帯域情報（図4のS804）の中から、最も大きな帯域情報（図4ではBand. C）を用いてコネクション設定の可否を判断する。つまり、送られてきたDTLスタック（Setupメッセージ）に記述されている要求帯域（Band. Req）によらずに、前述の、ネットワーク内に発生している呼が要求している最大帯域（Band. C）を呼の要求帯域と考えて、この値が、コネクションを設定しようとしている選択経路上のリンクやATMノード内の有効帯域（Band. Remain）よりも小さかった場合に、要求されたコネクションを設定可能と判断し、逆に大きい／等しい場合には、コネクションの設定を不可能と判断する（図4の処理S802）。

【0105】ピアグループ“201”内でのコネクション設定が可能と判断された場合には、選択された経路上での次のピアグループである“203”へのリンクである、ATMノード“104”とATMノード“109”の間のリンク上に要求されたコネクションが設定可能であるかを判断する。この時の判断アルゴリズムも、図4に示した方法と同様に実行することで実現できる。

【0106】この段階では、ソースノードにおける最初のルーティング処理の結果、送られてきたDTLスタック（Setupメッセージ）に関しての処理を行なっているので、コネクション設定可否判定に用いるアルゴリズムは、図4の“アルゴリズム-2”である。

【0107】[クラックバック処理S771]もし、図3の処理S710においてピアグループ“201”内でのコネクション設定が不可能と判断された場合には、クラックバック処理が実行される（図3の処理S771）。この時のクラックバック処理は、最初の、ソースノードによるルーティング処理の結果によるものである。クラックバックによってDTLスタック（Setupメッセージ）を差し戻されるATMノードは、必ずソースノード“101”である。

【0108】ソースノード“101”は、クラックバックされたDTLスタック（Setupメッセージ）を受け取ると、新たな経路選択処理を実行し、新たに転送するDTLスタック（Setupメッセージ）がクラックバック処理を実行した後のものであることを、そのクラックバック・インフォメーション・エレメント（Crackback Information Element）を用いて明示し、その作成したメッセージ（作成したDTLスタックを含む）を前述の方法と同様に選択した経路上に転送していく。

【0109】前述の処理S710におけるコネクション設定可否判定アルゴリズムでは、送られてきたDTLス

タック（Setupメッセージ）が、ソースノードにおける最初のルーティング処理の結果作成されたものであったので、図4の“アルゴリズム-2”が選択されていた。しかし、この処理の後には、送られてくるDTLスタック（Setupメッセージ）がクラックバック処理の後に作成されたものとなるので、各ATMノードで実行する判断アルゴリズムは、図4の“アルゴリズム-1”となる。

【0110】[図3の処理S702]図3の処理S710において、ピアグループ“201”内および、ピアグループ“203”へのリンクである、ATMノード“104”と“109”の間のリンク上に、要求されたコネクションが設定可能であると判断された場合に、ソースノードであるATMノード“101”で作成されたDTLスタック（Setupメッセージ）はATMノード“109”に転送され、この、ATMノード109において、新たに、ピアグループ“203”でのルーティング／シグナリング処理が実行される。ATMノード“109”は、受け取ったDTLスタック（Setupメッセージ）に、次の転送先のピアグループが“204”であると記述されているので、自ノード“109”からピアグループ“204”に接続するための、自ノードが属しているピアグループ“203”内のルーティング／シグナリング処理を実行する（図3の処理S702）。

【0111】図3においては、ATMノード109によるルーティング処理によって、経路[109-110-112-204]が選択された場合を想定している。

【0112】この時に、ATMノード109は、送られてきたDTLスタックの経路情報の中のピアグループ201内の経路情報を、新たに作成したピアグループ“203”内の経路情報に書き換えたDTLスタックを作成し、新たに選択した、ピアグループ“203”内の経路に沿って、そのDTLスタック（Setupメッセージ）を、順次転送していくことになる。

【0113】この時に、新たに選択された経路上に存在するリンクやATMノードは、先の場合と同様に、要求されている呼のQOS情報と自リンクやATMノードのリソース状況を比較し、要求されているQOS情報を満足することができる場合に、コネクション設定を可能と判断し、選択された経路上の次のATMノードにDTLスタック（Setupメッセージ）を転送する。

【0114】[図3の処理S720]次に、ATMノード“109”から送られてきたDTLスタック（Setupメッセージ）を受け取った各ATMノードにおいて、実際に要求された帯域が、自ノード内とノード間のリンク上に設定可能であるかどうかを判断する（図3の処理S720）。この時の、各ATMノードで具体的に実行されるアルゴリズムは、前述の場合と同様に図4に示したアルゴリズムである。よって、各ATMノードでは、送られてきたDTLスタック（Setupメッセー

ジ) がソースノードにおける最初のルーティング処理の結果作成されたものか、それとも何度かのクランクバック処理の後に作成されたものかを判断する。この結果、送られてきたDTLスタック (Setupメッセージ) が、ソースノードにおける最初のルーティング処理の結果、作成されたものであった場合には、図4の“アルゴリズム-2”を実行する。

【0115】つまり、送られてきたDTLスタック (Setupメッセージ) に記されている要求帯域 (Band, Req) がコネクションを設定しようとしているリンクやATMノード内の有効帯域 (Band, Remain) よりも小さかった場合に、要求されたコネクションを設定可能と判断し、逆に大きい／等しい場合には、コネクションの設定を不可能と判断する。

【0116】また、送られてきたDTLスタック (Setupメッセージ) がクランクバック処理の後のルーティング処理の結果作成されたものであった場合には、ネットワーク内に発生している呼が要求している帯域の中の最も大きな帯域と、コネクションを設定しようとしている選択経路上のリンクやATMノード内の有効帯域とを比較して、コネクションの設定可否の判断をする (図4の“アルゴリズム-1”)。

【0117】具体的には、本実施例のATMノード内に、ネットワーク内に発生している呼が要求している帯域情報を記憶する手段に記憶されている要求帯域情報

(図4のS804)の中から、最も大きな帯域情報 (図4では“Band, C”) を用いてコネクション設定の可否を判断する。つまり、送られてきたDTLスタック (Setupメッセージ) に記述されている呼の要求帯域 (“Band, Req”) によらずに、前述の、ネットワーク内に発生している呼の最大要求帯域を呼の要求帯域と考えて、この値が、コネクションを設定しようとしている選択経路上のリンクやATMノード内の有効帯域よりも小さかった場合に、要求されたコネクションを設定可能と判断し、逆に大きい／等しい場合には、コネクションの設定を不可能と判断する (図4の処理S802)。

【0118】ピアグループ“203”内でのコネクション設定が可能と判断された場合には、選択された経路上での次のピアグループである“204”へのリンクである、ATMノード“112”とATMノード“114”の間のリンク上に、要求されたコネクションが設定可能であるかどうかを判断する。この場合の判断アルゴリズムも、図4に示した方法と同様に実行することで実現できる。

【0119】[クランクバック処理S772] もし、図3の処理S720において、ピアグループ“203”内におけるコネクション設定が不可能と判断された場合には、クランクバック処理が実行される (図3の処理S772)。この時に実行されるクランクバック処理では、

必ず、ソースノード (ATMノード“101”) にDTLスタック (Setupメッセージ) が差し戻されるとは限らず、最初にピアグループ“203”内とピアグループ“203”と“204”の間の代替経路の存在が確認される場合がある (図3の処理S740)。

【0120】例えば、ピアグループ“203”内のATMノード“111”と“112”の間のリンク上へのコネクション設定が不可能と判断されてクランクバック処理が起動された場合であれば、そのピアグループ“203”内に [109-110-112-204] という代替経路が存在するので、DTLスタック (Setupメッセージ) をATMノード“109”まで差し戻し、再度、ピアグループ203内でのルーティング／シグナリング処理を実行する (図3の処理S781)。

【0121】これに対して、例えば、ピアグループ203とピアグループ204との間のように、経路が1本しかなく、この場合にピアグループ203とピアグループ204とのリンク上へのコネクション設定が不可能と判断された場合であれば、他にはピアグループ“204”に到達するための代替経路は存在しないことになる。

【0122】このような場合には、DTLスタック (Setupメッセージ) を、ピアグループ“203”におけるソースノードであるATMノード“109”よりも前のピアグループにおけるソースノード (この場合はATMノード“101”) にまで差し戻し、そこで再度、ATMノード“101”からピアグループ“204”へのルーティング／シグナリング処理を実行する (図3の処理S782)。

【0123】この時のルーティング／シグナリング処理においては、先にコネクション設定が不可能と判断されたリンクやATMノードは、そこで使用するトポロジ情報の中から削除する、もしくは、非常に大きな重みを与えてしまい、経路探索処理によって再度選択されないようにするなどして、新たな代替経路の選択処理を実行する方法が考えられる。

【0124】上記のような方法によって、クランクバック処理されたDTLスタック (Setupメッセージ) を受け取ったATMノード“109”またはATMノード“101”は、前述のような、新たな経路選択処理を実行し、クランクバック処理を実行した後のDTLスタック (Setupメッセージ) であることを、そのCrackback Information Elementを用いて明示したメッセージ (作成したDTLスタックを含む) を前述の方法と同様に選択した経路上に転送していくことになる。

【0125】このクランクバック処理の後に実行される、前述の処理S710、S720におけるコネクション設定可否判定アルゴリズムは、送られてきたDTLスタック (Setupメッセージ) がクランクバック処理の後に作成されたものであるため、図4の“アルゴリズム

ム-1”となる。

【0126】〔図3の処理S703〕図3の処理S720において、ピアグループ“203”内および、ピアグループ“204”へのリンクである、ATMノード“112”と“114”の間のリンク上に、要求されたコネクションが設定可能であると判断された場合に、ピアグループ“203”におけるソースノードであるATMノード“109”で作成されたDTLスタック（Setupメッセージ）はATMノード“114”に転送される。

【0127】そして、このATMノード“114”において、新たに、ピアグループ“204”内でのルーティング/シグナリング処理が実行される。ATMノード“114”は、自ノードが属しているピアグループが呼の要求する転送先のピアグループであるので、自ノードが属しているピアグループ内に呼の要求する実際の転送先ATMノードが存在することが認識できる。

【0128】本実施例では、転送先のATMノードを“116”としているので、ここでは、ATMノード“114”からATMノード“116”に接続するための、ピアグループ“204”内におけるルーティング/シグナリング処理を実行する（図3の処理S703）。

【0129】ATMノード“114”によるルーティング処理によって、経路〔114-116〕が選択されたとすると、ATMノード“114”は、送られてきたDTLスタック（Setupメッセージ）に記されているピアグループ“203”内の経路情報を、新たに作成したピアグループ“204”内の経路情報に書き換えたDTLスタック（Setupメッセージ）を作成し、上記のように、新たに選択した経路に沿って、そのDTLスタック（Setupメッセージ）を順次転送することになる。

【0130】この時に、新たに選択された経路上に存在するATMノードは、先の場合と同様に、要求されている呼のQOS情報と自ノードや隣接リンクのリソース状況を比較し、要求されているQOS情報を満足することができる場合に、コネクション設定を可能と判断し、選択された経路上の次のATMノードにDTLスタック（Setupメッセージ）を転送する。

【0131】〔図3の処理S730〕ATMノード“114”から送られてきたDTLスタック（Setupメッセージ）を受け取った各ATMノードにおいて、要求された帯域が、自ノード内とノード間のリンク上に設定可能であるかどうかを判断する（図3の処理S730）。この時の、各ATMノードで具体的に実行されるアルゴリズムは、前述の場合と同様に、図4に示したアルゴリズムである。

【0132】よって、各ATMノードでは、送られてきたDTLスタック（Setupメッセージ）がソースノードにおける最初のルーティング処理の結果、作成され

たものであった場合には、図4の“アルゴリズム-2”を実行する。

【0133】また、送られてきたDTLスタック（Setupメッセージ）がクランクバック処理の後のルーティング処理の結果作成されたものであった場合には、ネットワーク内に発生している呼が要求している帯域の中の、最も大きな帯域と要求された帯域を比較して、コネクションの設定可否の判断をする（図4の“アルゴリズム-1”）。

10 【0134】ピアグループ“204”内でのコネクション設定が可能と判断された場合には、最終的な転送先ATMノード“116”にDTLスタック（Setupメッセージ）が転送され、この呼接続要求に対する応答メッセージ（Connectメッセージ）を、呼接続要求を出した端末に対して送り返すことで、呼の接続処理が終了する。

【0135】〔クランクバック処理S773〕もし、図3の処理S730においてピアグループ“204”内でのコネクション設定が不可能と判断された場合には、再度、クランクバック処理が実行される（図3の処理S773）。この場合に実行されるクランクバック処理においても、必ず、ソースノードであるATMノード“101”にDTLスタック（Setupメッセージ）を差し戻すとは限らず、最初に、ピアグループ“204”内の代替経路の存在を確認する場合がある（図3の処理S750）。

【0136】この場合では、図1中のピアグループ“204”内のATMノード“114”と“116”の間のリンク上へのコネクション設定が不可能と判断されたことになるが、その代わりに、〔114-113-115-116〕という代替経路が存在する。よって、この時は、DTLスタック（Setupメッセージ）をATMノード“114”にまで差し戻し、再度、ピアグループ“203”内でのルーティング/シグナリング処理を実行する（図3の処理S783）。

【0137】ところが、例えば、一度ピアグループ“204”内でのクランクバック処理を実行した後に、再度、図3の処理S703、S730を実行して、その結果が、やはりコネクションの設定不可能であった場合には、ATMノード“114”からATMノード“116”への代替経路は、もう存在しないことになる。

【0138】この時には、クランクバック処理の実行によってDTLスタック（Setupメッセージ）をATMノード“109”まで差し戻すことになる。

【0139】この時に、すぐに経路選択処理（図3の処理S702）を実行することもできるが、そのような方法だけではなく、一度、ATMノード“109”からATMノード“116”への代替経路が存在するかどうかを確認してから経路選択処理に移ることも可能である。

50 【0140】図3の例では、後者の方法を示している。

この時、すでにATMノード“114”からATMノード“116”への経路にはコネクションを設定できないことがわかっているため、再度ATMノード“109”において経路選択処理を行っても、新たなコネクションが設定可能な経路を発見することはできない。

【0141】よって、この場合には、DTLスタック（Setupメッセージ）を、ソースノードであるATMノード101まで差し戻して、再度、ATMノード“101”からピアグループ“204”に対しての経路探索処理を実行する（図3の処理S784）。

【0142】この経路探索処理の後に、再度、選択経路上へのコネクションの設定可否を判定しながら、コネクションの設定を試みる。このような処理の後に実行されるコネクション設定可否の判断においては、送られてくるDTLスタック（Setupメッセージ）がクラックバック処理の後のものとなるので、図4の“アルゴリズム-1”が用いられることになる。

【0143】このように、本発明のATMノードにおけるコネクション設定可否の判断に用いる帯域情報を、そのコネクション設定可否の判断を要求するDTLスタック（Setupメッセージ）がクラックバック処理の前に作成されたものであるのか、後で作成されたものであるのかによって変更することで、クラックバック処理による効果をネットワーク内に発生している呼の要求する帯域情報に関係なく提供することができるようになる。

【0144】特に、クラックバック処理の後に作成されたDTLスタック（Setupメッセージ）によって起動されるコネクション設定可否の判定を行なう際に、コネクションを設定しようとしているATMノード内やリンク上の有効帯域と、ネットワーク内に発生している呼が要求している帯域の中の最大帯域の値とを比較してコネクション設定の可否を判断することによって、ネットワーク中に発生する全ての呼がクラックバック処理による効果を均等に享受できるようになる。

【0145】[ATMノードの内部構成] 図7に、本実施例に示したATMノードの内部機能構成の一例をブロック図で示す。図7に示した例では、実際のATMセルの転送/交換処理を実行するためのATMインタフェース処理部1104、1106と、ATMスイッチ1105が存在する。

【0146】また、一連のルーティング処理、DTLスタック（Setupメッセージ）の転送/受信処理、コネクション設定可否判定処理などのPNNIルーティング/シグナリング処理を実行するPNNIルーティング/シグナリング処理実行部1103と、その中のクラックバック処理のためのDTLスタック（Setupメッセージ）の記憶や、クラックバック処理が発生した際に差し戻すATMノードの決定や、代替経路の存在確認処理等を実行するクラックバック処理実行部1102がある。

【0147】さらに、本実施例に示すATMノードには、PNNIルーティング/シグナリング処理実行部1103が受け取ったDTLスタック（Setupメッセージ）が、クラックバック処理の後のものであるか、否かを判断して、その結果から、コネクション設定可否の判断に用いるアルゴリズムを決定/選択する、アルゴリズム選択処理部1101がある。

【0148】これらの処理部によって、実際のシグナリング処理を実行する場合の例として、図3の処理S702からS720、S740までの処理を実行する際の、ATMノード“109”における、各機能の具体的な処理手順を以下に示す。

【0149】図7に示す如き構成を持つATMノード“109”では、ATMノード“104”から、ATMインタフェース処理部1106を介してDTLスタック（Setupメッセージ）を受け取る。そして、このDTLスタックを受け取った当該ATMノード“109”は、その受け取ったDTLスタック（Setupメッセージ）をPNNIルーティング/シグナリング処理実行部1103に転送する。

【0150】また、PNNIルーティング/シグナリング処理実行部1103から、受け取ったDTLスタック（Setupメッセージ）をアルゴリズム選択処理部1101に転送し、そこで、受け取ったDTLスタック（Setupメッセージ）がクラックバック処理されたものであるのか否かを判断する。

【0151】その結果、アルゴリズム選択処理部1101は、このDTLスタック（Setupメッセージ）がクラックバック処理されたものでなかった場合には、ATMノード“109”内でのコネクションの設定可否の判断に図4の“アルゴリズム-2”を用いるように指示し、DTLスタック（Setupメッセージ）がクラックバック処理された後に作成されたものであった場合には、ATMノード“109”内でのコネクションの設定可否の判断において、図4のステップS802における“アルゴリズム-1”を用いるように指示する。

【0152】この時の、アルゴリズム選択処理部1101での、送られてきたDTLスタック（Setupメッセージ）の種別の判断においては、受け取ったDTLスタック（Setupメッセージ）中のCrankback Information Elementの記述やCause Fieldを参照することによって可能である。

【0153】上記の結果、選択されたアルゴリズムを用いて、まず、要求されている帯域がATMノード“109”内に設定可能であるか否かを判断し、その結果、設定可能であると判断された場合には、ピアグループ“203”内でのコネクション設定経路の探索処理を実行する。

【0154】この時には、ATMノード“109”が記

憶しているネットワークの仮想的トポロジー情報と、受け取ったDTLスタック (Setupメッセージ) に記述されている全体の経路情報と宛先情報により、自ノードが属するピアグループ“203”内における、経路探索処理を実行することになる。

【0155】図3に示したように、ここで[109-110-112-204]という経路が選択されたとすると、まず、ATMノード“109”とATMノード“110”との間のリンク上に、コネクションが設定可能であるかどうかを、先に選択されたアルゴリズムによって評価する。そして、その評価の結果、設定可能と判断された場合には、新たにDTLスタック (Setupメッセージ) を書き直して、選択した経路上の次のATMノードに、書き直したDTLスタック (Setupメッセージ) を転送する。逆に、設定不可能と判断された場合にはクラックバック処理が起動される。

【0156】ATMノード“109”によって選択された経路上のATMノード、例えばATMノード“110”は、先の場合と同様に受け取ったDTLスタック (Setupメッセージ) がクラックバック処理されたものかどうかを調べ、使用するコネクション設定可否の判断アルゴリズムを選択する。

【0157】ここで、もし、コネクション設定が不可能であると判断されたならば、そのDTLスタック (Setupメッセージ) をクラックバック処理実行部1102に転送し、どのATMノードまでDTLスタック (Setupメッセージ) を差し戻すのかを決定する。

【0158】また、この時のDTLスタック (Setupメッセージ) の種類や、DTLスタック (Setupメッセージ) に記述されている内容を記憶するとともに、コネクションが設定不可能と判断された理由などを、新たな情報としてDTLスタック (Setupメッセージ) に付加してクラックバック処理 (DTLスタック (Setupメッセージ) の差し戻し処理) を実行することになる。

【0159】以上、PNNIプロトコルを実装したATMノードによって構成されるネットワークにおいて、シグナリング処理がクラックバック後の処理であるのか、クラックバック前の処理であるのかによって、コネクション設定可否を決定するアルゴリズムを変更することで、クラックバック処理による効果の受け方が、コネクションの種類によって不公平にならないようにできるATMノードを提供できる。従来、PNNIプロトコルにおけるクラックバック処理による効果の受け方が、コネクションの種類によって違ってしまうことによる不公平があった。特に、要求帯域の異なる呼が存在するマルチコネクションの環境において、この不公平性が顕著になる。そこで、本発明では、ATMノードにおけるシグナリング処理時に、そのシグナリング処理がクラックバック後であるのか否かにより、コネクション設定可否を決

定するアルゴリズムを変更することで、前記の不公平性を是正するようにした。

【0160】なお、本発明は上述の例に限定されるものではなく、種々変形して実施可能である。

【0161】

【発明の効果】以上説明したように、本発明によるATMノードによればPNNIインタフェースで接続されたATMノードによって構成されるATMネットワークにおける呼設定処理において、マルチコネクションの環境においても、PNNI処理の一つであるクラックバックの効果も、全ての種類のコネクションが均等に享受できるようになる利点が得られる。

【図面の簡単な説明】

【図1】本発明を説明するための図であって、本発明のATMノードを用いたネットワークのトポロジーの一例を示す図。

【図2】本発明を説明するための図であって、本発明のATMノードが認識するネットワークのトポロジーの一例を示す図。

【図3】本発明を説明するための図であって、本発明のATMノードで実行するシグナリング処理の手順を示すフローチャートの一例を示す図。

【図4】本発明を説明するための図であって、本発明のATMノードで実行するシグナリング処理の手順の、特にコネクション設定可否を決定するプロセスの詳細な一例を示す図。

【図5】本発明を説明するための図であって、クラックバック・インフォメーション・エレメント (Crank back information Element) の一例を示す図。

【図6】本発明を説明するための図であって、コース・フィールド (Cause Field) の一例を示す図。

【図7】本発明を説明するための図であって、本発明のATMノードの内部構成の一例を示すブロック図。

【図8】PNNIにおけるネットワークトポロジー認識方式の概念を示す図。

【図9】従来のルーティング方式が用いていたトポロジー情報の一例を示す図。

【図10】PNNIプロトコルにおけるトポロジー認識方式の概念を説明するための図。

【図11】PNNIプロトコルにおけるDTLスタックを用いたシグナリング手順の例を説明するための図。

【図12】PNNIプロトコルにおけるDTLスタックを用いたシグナリング手順の例を説明するための図。

【図13】PNNIプロトコルにおけるクラックバック処理手順の例を説明するための図。

【符号の説明】

101~116…ATMスイッチノード (ATMノード)

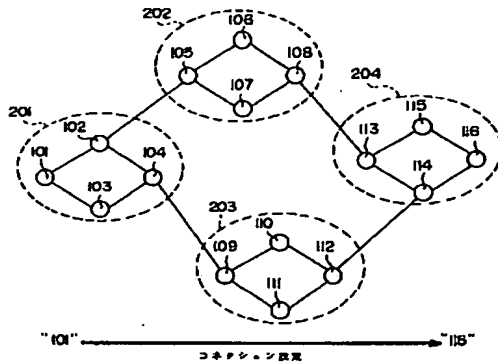
201~204...ピアグループ

1101...アルゴリズム選択処理部

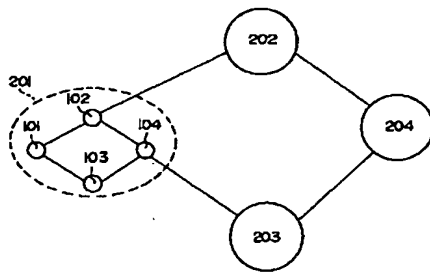
1102...クラックバック処理実行部

1103...PNNIルーティング/シグナリング処理実

〔図1〕



〔図2〕



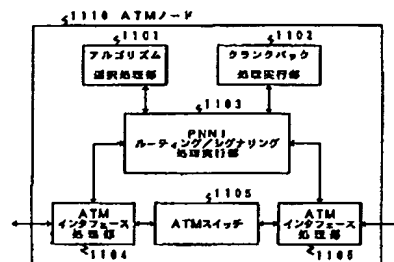
行部

1104, 1106...ATMインタフェース処理部

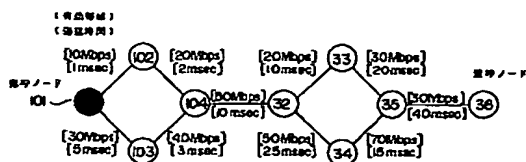
1105...ATMスイッチ

1110...ATMノード。

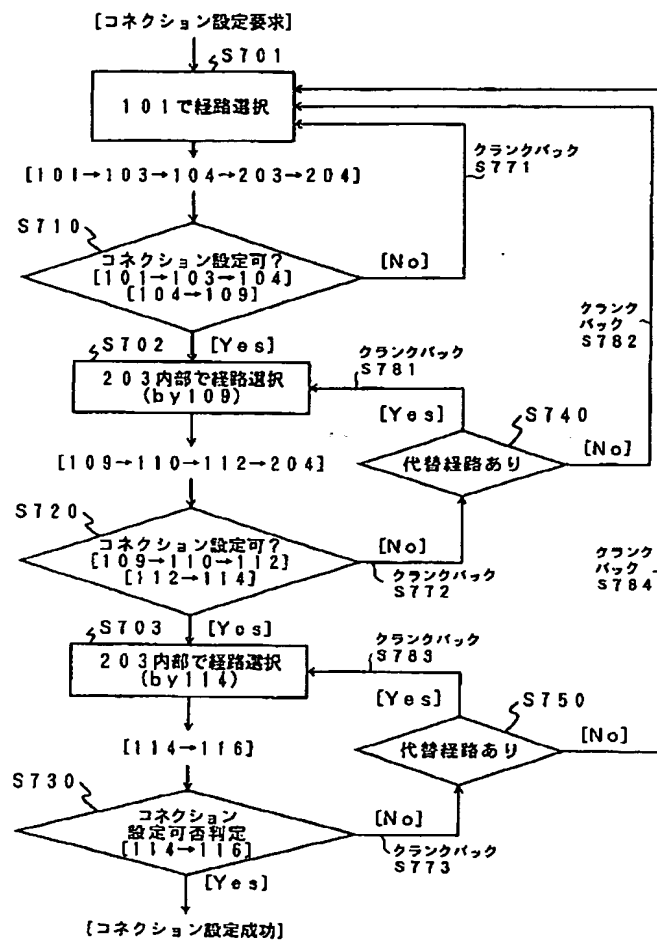
〔図7〕



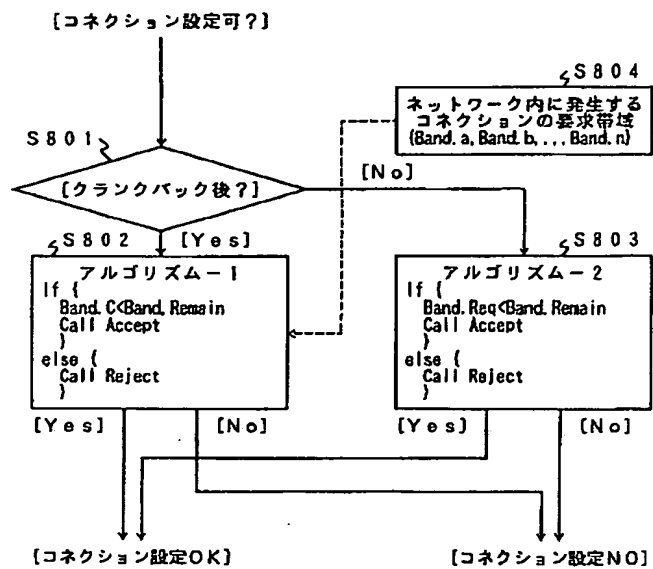
〔図9〕



【図3】



【図4】



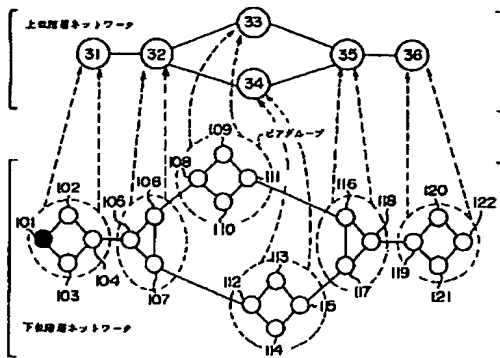
【図6】

Crashback cause (octet 9.1)

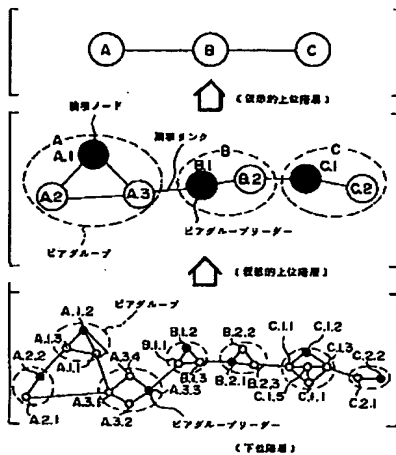
Bits 8 7 6 5 4 3 2 1	Number	Meaning	Diagnostics
0000 0010	2	transit network unreachable	Note 2
0000 0011	3	destination unreachable	
0010 0011	35	requested VPC/VCI not available	
0010 0101	37	user cell rate not available	
0010 1101	45	no VPC/VCI available	
0010 1111	47	resource unavailable, unspecified	Note 3
0011 0001	49	Quality of Service unavailable	
1000 0000	128	next node unreachable	Note 1
1010 0000	160	DTL Transit not by node ID	Note 4
1010 0001	161	port ID in DTL not available	

Crashback cause diagnostics

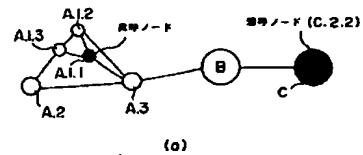
〔図8〕



〔図10〕



〔図11〕



〔作成されるDTLスタック〕

A.1.1	A.1.2	A.1.3
A.1-A2-A3		
A-B-C		

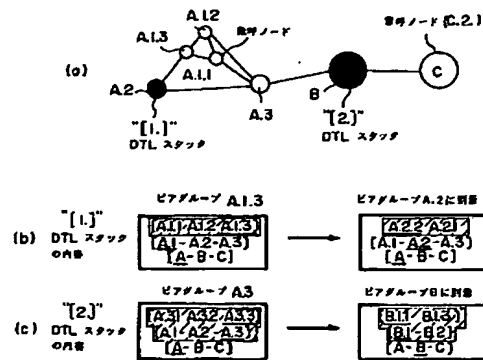
〔A.1内経路参照〕

〔A内経路参照〕

〔最上位階層参照〕

(b)

[図12]



[図13]

